Tổng hợp tiếng nói sử dụng Mô hình Markov ẩn trên nền DSP

- **Vũ Trương Minh Nhật**
- **Nguyễn Hiếu Bình**
- **Phạm Minh Nhật**
- **Huỳnh Hữu Thuận**
- **Bùi Trọng Tú**
- **Vũ Hải Quân**

Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

(Bài nhận ngày 20 tháng 3 năm 2013, nhận đăng ngày 20 tháng 3 năm 2014)

TÓM TẮT

Tổng hợp tiếng nói bằng phương pháp thống kê Mô hình Markov ẩn (HMM) đã trở nên phổ biến trong những năm gần đây và hầu hết được thực hiện trên máy tính cá nhân (PC). Để có thể triển khai trên thực tế hiệu quả thì hệ thống phải được thực hiện trên một hệ thống nhúng cụ thể. Bài báo này trình bày phương pháp tổng hợp tiếng nói trên nền DSP (digital signal processing). Bằng các phương pháp tối ưu hóa, chất lượng tiếng nói tạo ra và thời gian xử lý trên board DSP có chất lượng tương đương với tổng hợp tiếng nói trên máy tính cá nhân.

Từ khóa: *Chuyển từ văn bản sang tiếng nói tiếng Việt, mô hình Markov ẩn, xử lý tín hiệu số*

MỞ ĐẦU

Phương pháp thống kê số trong tổng hợp tiếng nói dựa trên Mô hình Markov ẩn (HMM) đã trở nên phổ biến trong những năm gần đây. Trong phương pháp này các mô hình HMMs được trích xuất từ tiếng nói tự nhiên nên ta có thể xây dựng được mô hình HMM cho bất kỳ ngôn ngữ nào. Sau đó tiếng nói được tổng hợp từ chính các mô hình HMM này. Việc sử dụng HMM giúp ta tổng hợp được nhiều giọng nói với những cảm xúc khác nhau mà không cần một cơ sở dữ liệu lớn thông qua điều chỉnh các thông số của mô hình như là thông số phổ, thông số kích thích, hay độ dài âm. Trong phạm vi bài báo nhóm tác giả sử dụng mô hình HMMs cho tiếng nói phương nam, với tần số lấy mẫu là 48Khz, và được huấn luyện bởi Phòng thí nghiệm Trí tuệ nhân tạo (AILab) Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM).

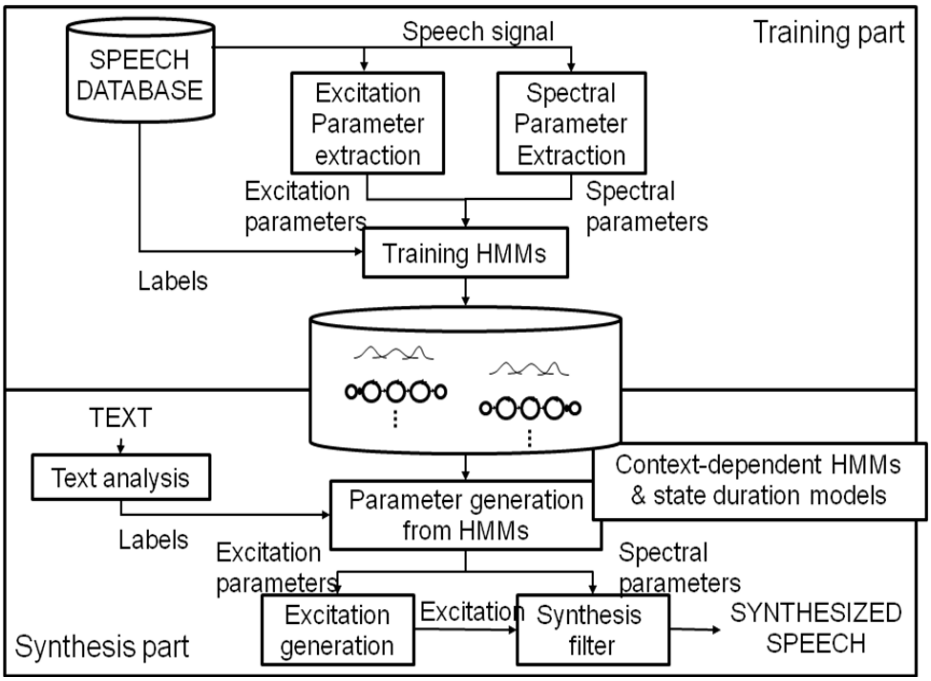
Bài báo thực hiện hts_engine trên board DSP(DM642 EVM), hts_engine là phần mềm mã nguồn mở của Học viện Kỹ thuật Công nghệ Tokyo, dùng để tạo ra mã PCM từ các mô hình Markov ẩn với đầu vào là văn bản.

Board DM642EVM có bộ xử lý trung tâm là chip DM642, chip DM642 dựa trên chip xử lý C64x thuộc họ C600. Chip DM642 có 16Kbyte cache L1P(level-1 program), 16Kbyte cache L1D(level-1 data), 256Kbyte có thể làm bộ nhớ trong(IRAM) mức 2 hoặc cache mức 2. Board DM642EVM có 64 Mbyte SDRAM và 4 Mbyte bộ nhớ flash, bộ giải mã âm thanh AIC23B stereo codec, giao tiếp máy tính bằng cổng JTAG. Mã PCM sau khi được tổng hợp sẻ được chuyển qua AIC23B để chuyển thành âm thanh.

PHƯƠNG PHÁP

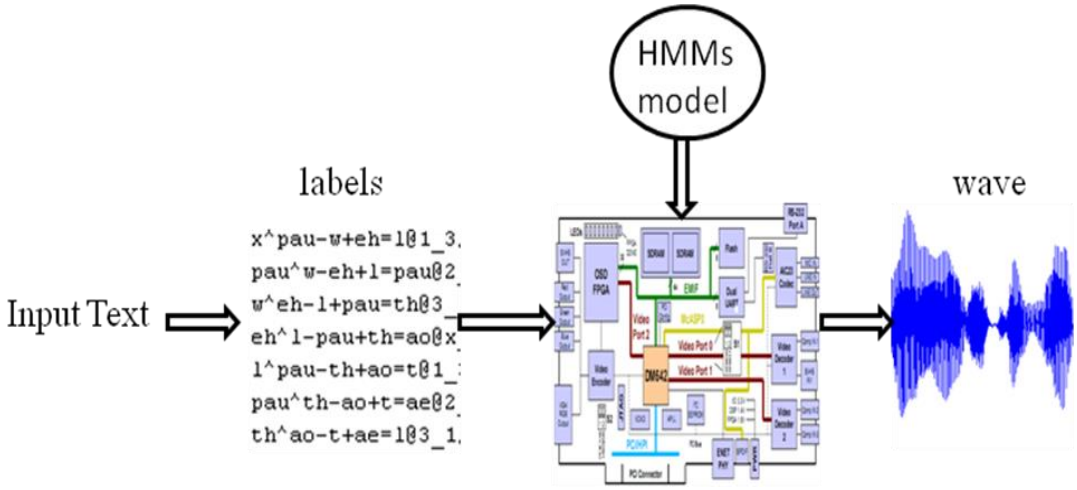
Tạo tham số

Hình 1 cho ta cái nhìn tổng quan của hệ thống [4]. Trong phần huấn luyện, các thông số kích thích, thông số phổ được trích xuất và được mô hình hóa. Trong phần tổng hợp các mô hình HMM được chọn lọc và ghép nối dựa trên đoạn văn bản cần được tổng hợp. Sau đó các thông số kích thích và thông số phổ được tạo ra bởi thuật toán được trình bày trong [1-3].



Hình 1. Tổng quan hệ thống HTS.

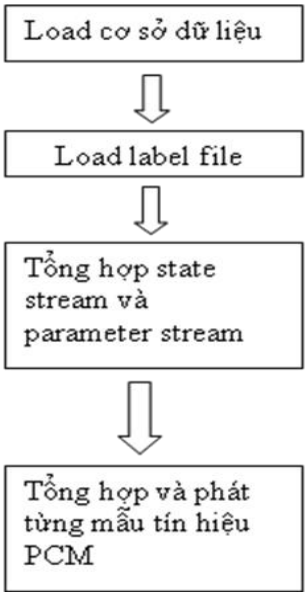
Thực hiện trên DSP



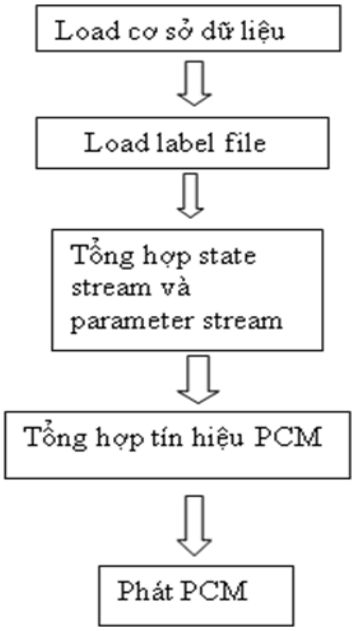
Hình 2. Sơ đồ thực hiện hts_engine trên DSP board(DM642EVM).

Mã chương trình được đưa vào IRAM, dữ liệu yêu cầu kích thước bộ nhớ lớn nên được đặt trên SDRAM. Cơ sở dữ liệu load từ máy tính xuống SDRAM. Cơ sở dữ liệu chứa các tham số của các mô hình Markov ẩn đã được huấn luyện. Trước khi được tối ưu hóa, để HTS engine có thể hoạt động trên board DM642 EVM, quá trình hoạt động của chương trình cần thay đổi so với chương trình gốc. Nguyên nhân của việc phải thay đổi là do thời gian tổng hợp từng mẫu PCM trên board DM642 EVM lớn hơn chu kì lấy mẫu của tín hiệu.

Tần số lấy mẫu được thiết lập bằng 48 Khz, chu kì lấy mẫu là $\frac{1}{48000}$ s. Mã chương trình gốc phát từng mẫu PCM ngay sau khi nó được tổng hợp. Với mã chương trình gốc, khi hoạt động trên máy tính, thời gian để tổng hợp một mẫu PCM ít hơn $\frac{1}{48000}$ s, nhưng khi chạy trên DM642 EVM thời gian tổng hợp nhiều hơn $\frac{1}{48000}$ s. Điều đó giải thích tại sao chương trình ban đầu có thể chạy được trên máy tính nhưng không thể cho ra kết quả như ý trên board DM642 EVM. Để có được kết quả như ý, ta cho chương trình phát tín hiệu PCM sau khi tất cả các mẫu PCM đã được tổng hợp và được lưu vào một bộ đệm. Theo sơ đồ sau:



Hình 3. Lưu đồ thực hiện HTS trên PC



Hình 4. Lưu đồ thực hiện HTS trên board DSP

Kết quả: tín hiệu PCM tạo thành trên Board DM642 EVM giống với tín hiệu PCM tạo thành trên máy tính. Tiếng nói phát ra từ board DM642 EVM giống với tiếng nói tổng hợp được trên máy tính. Thời gian tổng hợp ra tín hiệu PCM trên board rất lâu so với thời gian tổng hợp trên máy tính. Sử dụng một label file tiếng Anh có độ dài tương đương 2s thì: thời gian thực hiện chương trình từ lúc bắt đầu đến lúc phát chuỗi PCM là 1 phút 35 giây; thời gian tổng hợp state stream, parameter stream, tổng hợp tín hiệu PCM là 42 giây.

Chương trình chạy chậm do ba nguyên nhân sau: Dữ liệu đặt trong SDRAM có tốc độ truy xuất chậm; tập lệnh của chip DM642 không hỗ trợ các phép toán giữa các số kiểu dấu chấm động (floating point); cơ sở dữ liệu được tải từ máy tính xuống SDRAM của board bằng cổng JTAG có tốc độ chậm.

Tối ưu hóa

Khi hoạt động độc lập, cơ sở dữ liệu sẽ được đặt trên bộ nhớ của board, không tốn thời gian tải từ máy tính xuống, xem như nguyên nhân thứ ba đã được giải quyết. Vì vậy hai nguyên nhân đầu cần được ưu tiên khắc phục. Để tăng tốc độ truy xuất dữ liệu, một phần IRAM còn trống được chuyển thành Level-2 cache 32 KB 4-way, dữ liệu trên SDRAM sẽ được map vào Level-2 cache. Bên cạnh đó, bộ DMA được sử dụng để truyền nhanh một số dữ liệu [5, 6].

Để tối ưu hóa tốc độ xử lý dữ liệu, cần thay các phép toán giữa các dữ liệu kiểu dấu chấm động thành các phép toán giữa các dữ liệu dấu chấm tĩnh. Quá trình tốn nhiều thời gian là quá trình tổng hợp các mẫu PCM, vì vậy ta áp dụng phương pháp trên ở quá trình này: Sau khi tổng hợp chuỗi trạng thái (state stream) và chuỗi tham số (parameter stream), chuyển các dữ liệu dạng dấu chấm động (floating-point) thành dạng dấu chấm tĩnh (fixed-point), định dạng Q20 (20 bit biểu diễn phần sau dấu chấm). Ở phần tổng hợp tín hiệu PCM, các phép toán của dấu chấm động được thay thế bằng các phép toán của dấu chấm tĩnh.

Cơ bản của việc thay thế kiểu dấu chấm động bằng dấu chấm tĩnh định dạng là tìm cách biểu diễn số thực bằng một số nguyên. Công thức biến đổi một số thực thành số nguyên đại diện cho số thực dấu chấm tĩnh (định dạng Q20) như sau::

$$y = x \times 2^{20}$$

x: số thực cần biểu diễn

y: số nguyên dùng để biểu diễn số thực

Phép nhân với 2^{20} tương đương phép dịch trái 20 bits trong tập lệnh của chip DM642

Ví dụ: hai số thực 1 và 0.5 dấu chấm tĩnh Q20 sẽ được biến đổi thành số nguyên như sau:

1→100...000(20 chữ số 0)

0.5→10...000(19 chữ số 0)

Để thực hiện tính toán, việc xây dựng thư viện mới cho các phép toán là cần thiết. Phép cộng và phép trừ giữa hai số thực kiểu dấu chấm tĩnh giống như phép cộng và trừ giữa hai số nguyên, vì vậy đó cũng là phép cộng và trừ trong tập lệnh của chip DM642. Phép nhân giữa hai số thực dấu chấm tĩnh Q20 khi biểu diễn như số nguyên được thực hiện tuần tự bởi hai phép toán trong tập lệnh như sau:

$$x_1 \times x_2 \rightarrow y_1 \times y_2 : 2^{20}$$

x_1, x_2: hai số thực được nhân với nhau

y_1, y_2: hai số nguyên dùng để biểu diễn số hai số thực

Phép toán đầu tiên là phép nhân giữa hai số nguyên. Phép toán thứ hai là chia cho 2^{20}, tương đương phép dịch phải 20 bits. Tương tự, phép chia giữa hai số thức dấu chấm tĩnh được thực hiện bởi hai phép toán trong tập lệnh của chip DM642:

$$x_1 \times x_2 \rightarrow y_1 \times 2^{20} : y_2$$

x_1, x_2: hai số thực được nhân với nhau

y_1, y_2: hai số nguyên dùng để biểu diễn số hai số thực

Phép toán đầu tiên là phép nhân một số nguyên với 2^{20}, tương đương phép dịch trái 20 bits. Phép toán thứ hai là phép chia giữa hai số nguyên.

Lưu ý, để kết quả của phép nhân và phép chia hai số thực có sai số nhỏ nhất, thứ tự phép toán phải được thực hiện như trên, không nên thay đổi thứ tự thực hiện các phép toán.

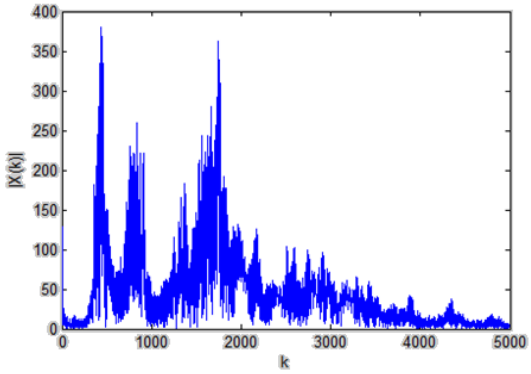
Việc xây dựng các hàm số học khác được thực hiện dựa trên bốn phép toán cơ bản trên và phép dịch bit. Sau khi áp dụng phương pháp chuyển kiểu dữ liệu dấu chấm động thành dấu chấm tĩnh và sử dụng thư viện mới được xây dựng, tốc độ tổng hợp tín hiệu PCM đã được cải thiện đáng kể. Sử dụng cùng một label file có độ dài tương đương 2s và 9s được kết quả như sau:

Bảng 1. Thời gian tổng hợp tín hiệu PCM

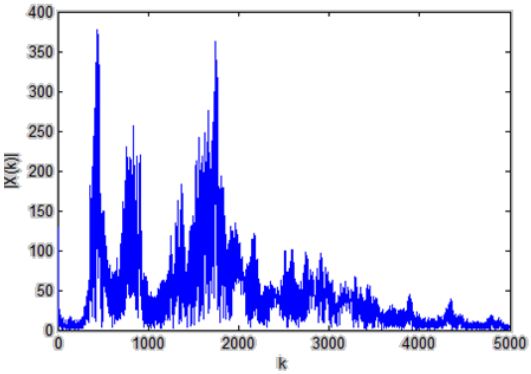
Độ dài label	2s	9s
Không cache và không DMA floating pointed.	42s	390s
Có cache và DMA cho floating pointed.	16s	63s
Không cache và không DMA cho fixed pointed.	31s	255s
Có cache và có DMA cho fixed pointed.	1s	4s

KẾT QUẢ

Dùng biến đổi Fourier rời rạc để lấy tần phổ của tín hiệu. So sánh tần phổ của tín hiệu PCM khi sử dụng dữ liệu kiểu dấm chấm tĩnh và và tần phổ của tín hiệu PCM ban đầu, ta thấy tần phổ có dạng rất giống nhau.



Hình 6. Phổ cho fixed point



Hình 7. Phổ cho floating point

Để so sánh sự giống nhau của hai tín hiệu X và Y, người ta thường sử dụng khái niệm hệ số tương quan Pearson trong toán học thống kê. Hệ số tương quan Pearson [7] được tính theo công thức:

$$\rho_{XY} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Để so sánh sự giống nhau của hai tín hiệu X và Y, hệ số tương quan được tính cụ thể như sau [7]:

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}}$$

Nếu hệ số tương quan có giá trị nằm trong khoảng 0.9 tới 1.0 nghĩa là hai tín hiệu rất giống nhau. Áp dụng cộng thức trên để tính hệ số tương quan tần phổ tín hiệu tạo thành từ chương trình gốc và tín hiệu tạo thành sau khi áp dụng các phép toán trên dấu chấm tĩnh, ta được kết quả 0.995. Vậy, hai tín hiệu tạo thành có dạng giống nhau. Trong thực tế, nếu cảm nhận bằng tai sẽ không thấy có sự khác biệt giữa hai tiếng nói trên.

KẾT LUẬN

Tổng hợp tiếng nói dùng mô hình Markov ẩn trên board DSP cho chất lượng tương đương trên máy PC. Nhưng nhóm tác giả vẫn chưa khai thác hết tính năng và những công cụ hỗ trợ tính toán trên board DSP. Do đó chất lượng tiếng nói và tốc độ xử lý trên board DSP vẫn còn có thể được cải thiện.

Performing Text – To – Speech based Hidden Markov Model on Digital Signal Processing platform

- **Vu Truong Minh Nhat**
- **Nguyen Hieu Binh**
- **Pham Minh Nhat**
- **Huynh Huu Thuan**
- **Bui Trong Tu**
- **Vu Hai Quan**

University of Science, VNU-HCM

ABSTRACT

Text To Speech (TTS) using Hidden Markov Model (HMM) has become popular in recent years. However, because most of such systems were implemented on personal computers (PCs), it is difficult to offer these systems to real applications. In this paper, we present a hardware implementation of TTS based on DSP architecture, which is applicable for real applications. By optimizing hardware architecture, the quality of the DSP-based synthesized speech is nearly identical to that synthesized on PCs.

Keywords: *Vietnamese Text to speech, hidden Markov model, digital signal processing*

TÀI LIỆU THAM KHẢO

[1]. K. Tokuda, T. Masuko, T. Yamada, T. Kobayashi, S. Imai, An algorithm for speech parameter generation from continuous mixture HMM with dynamic features, Proceeding of: Fourth European Conference on Speech Communication and Technology, EUROSPEECH 1995, Madrid, Spain (1995).

[2]. K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, T. Kitamura, Speech parameter generation algorithms for HMM-based speech synthesis, Proc. of ICASSP 2000, 3, 1315-1318 (2008).

[3]. T. Fukuda, K. Tokuda, T. Kobayashi, S. Imai, An adaptive algorithm for mel-ceptral analysis of speech, Proc. of ICASSP'92, 1, 137-1401 (1992).

[4]. K. Tokuda, H. Zen, Alan W. Black, An HMM-based speech synthesis applied to English, Proceedings of IEEE Workshop in Speech Synthesis (2002).

[5]. TMS320C6000 Chip Support Library API Reference Guide, Literature Number: SPRU401J, TEXAS INSTRUMENT (2004).

[6]. TMS320C6000 DSP cache User's Guide; Literature Number: SPRU656A; TEXAS INSTRUMENT (2003).

[7]. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient